

コードでAWS環境を管理しよう！ AWS Cloud Development Kit (CDK)ハンズオン

内田 大樹

Amazon Web Services Japan
Solution Architect

自己紹介



内田 大樹 (うちだ ひろき)

インダストリソリューション部
ソリューションアーキテクト

前職までの経験

通信会社でアプリケーション開発、監視、インフラ構築する部署に所属。

好きなAWSのサービス

AWS Command Line Interface (CLI)
AWS CDK



@nikuyoshi

内容についての注意点

- 本資料では2020年7月29日収録時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。資料作成には十分注意しておりますが、資料とAWS公式ウェブサイトとで記載内容に相違があった場合、AWS公式ウェブサイトの記載を優先させていただきます。
- マネージメントコンソールについても、収録時点のものとなります。差異がある場合がございますので、ご注意ください。
- ハンズオンでは、AWS の各種サービスの利用、リソースの作成を行います。無料枠を超えるコースもございますが、その場合、ご利用料金が発生することをあらかじめご理解ください。
- 学習後のリソースの削除についても、お客様の責任でご実施いただくようお願いいたします。

本コースのゴール/前提条件、前提知識

- 本コースのゴール

- コードでAWS環境を管理する方法を理解すること
- AWS CDK (TypeScript) を利用したAWS環境の構築方法を学ぶこと

※他のプログラミングに関しては動画ではなく、手順書を基に実施してください。

- 本コースの前提条件、前提知識

- AWSアカウントをお持ちであること
- ハンズオンの作業が同一AWSアカウントの他のリソースに影響がでる場合がある
- ハンズオン用にAWSアカウントを取得していただくことを推奨する
- 以下のAWSのサービスの基礎知識を持っていること。必須ではない。
 - AWS Lambda
 - Amazon API Gateway
 - AWS CloudFormation

本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. AWS CDKのハンズオン ①新しいプロジェクトの作成
 3. AWS CDKのハンズオン ②Hello, CDK!
 4. AWS CDKのハンズオン ③Costructの補足
3. 本コースのまとめ、リソース削除

本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備

2. AWS CDKの紹介とハンズオン

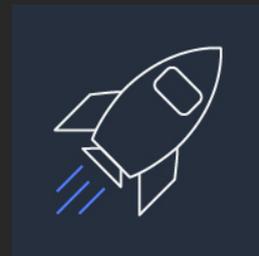
1. AWS CDKの概要
2. AWS CDKのハンズオン ①新しいプロジェクトの作成
3. AWS CDKのハンズオン ②Hello, CDK!
4. AWS CDKのハンズオン ③Costructの補足

3. 本コースのまとめ、リソース削除

ハンズオンで利用するサービス紹介 と事前準備

AWS Lambda の概要

- **サーバーのプロビジョニング/管理なしでプログラムを実行できるサービス**
- コードの実行やスケーリングに必要なことは、Lambda 側で実施するので、開発者の方はコードを書くことにより集中できる
- リクエストベースの料金体系



実行回数
(無料枠あり)

+

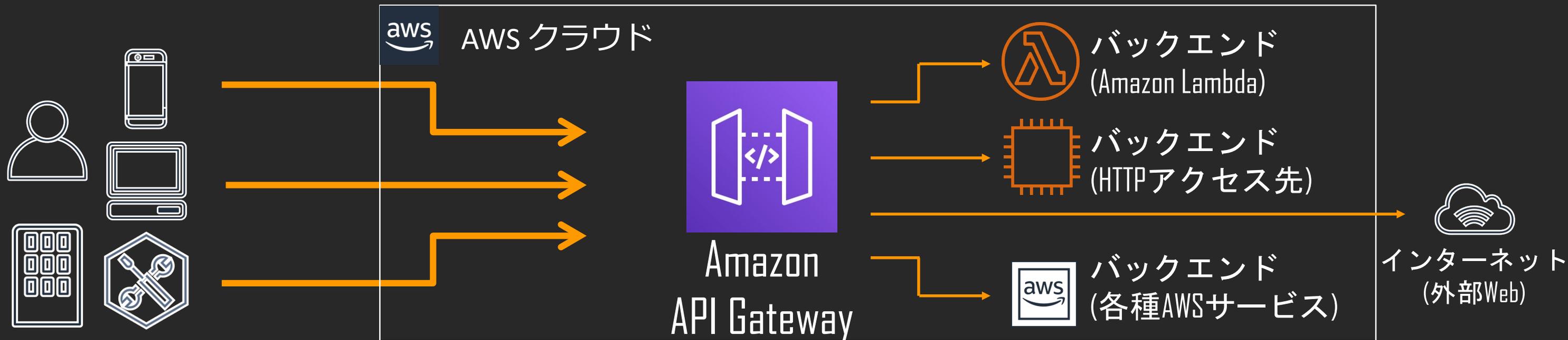


実行時間
(単価は確保したメモリによる)
(無料枠あり)

Amazon API Gateway の概要

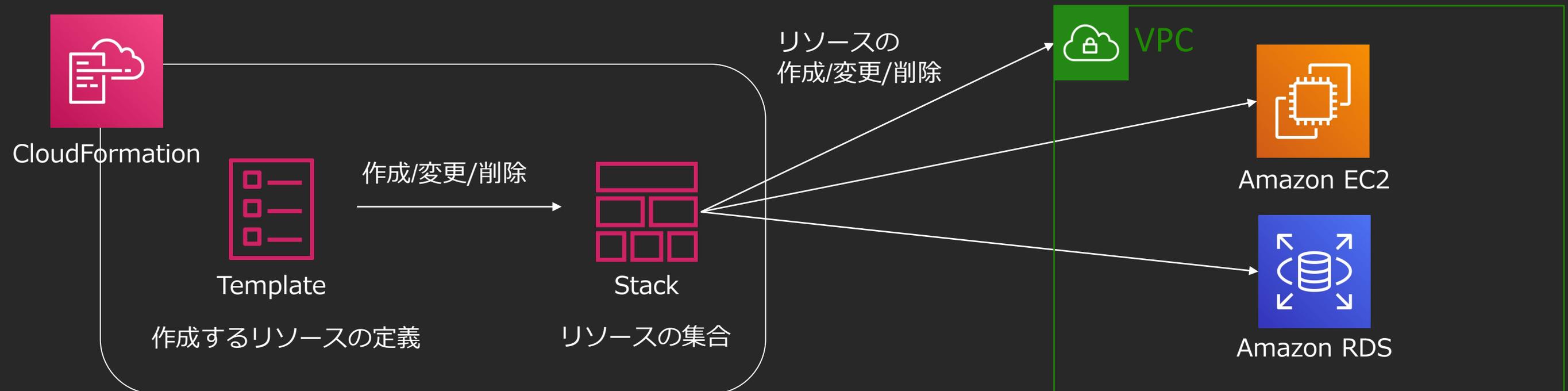
APIの作成、公開、保守、モニタリング、保護を簡単に行えるサービス

APIは、アプリケーションがバックエンドサービスからのデータ、ビジネスロジック、機能にアクセスするための「フロントドア」として機能する



AWS CloudFormation

- シンプルなテキストファイル（テンプレート）を使用して、リージョンとアカウントでアプリケーションに必要なとされる**AWSリソースをプロビジョニングできるサービス**
- CDKを利用することで、CloudFormationのテンプレートを自動生成できる
- プロビジョニングされたAWSリソースに対する変更/削除も可能で、ソフトウェアに対して行うのと同じ方法でAWSインフラを管理できる
- 追加料金なし（プロビジョニングされたAWSリソース分の料金のみ発生）



ここから事前準備を始めます

CDK 開発環境の準備

1. 前提条件

- AWS CLI
- Node.js 10.3.0 以上の環境を用意
 - Windows: MSIインストーラ / Mac: homebrew & nodebrew
- AdministratorAccess権限を持ったIAMユーザー

2. CDKの導入

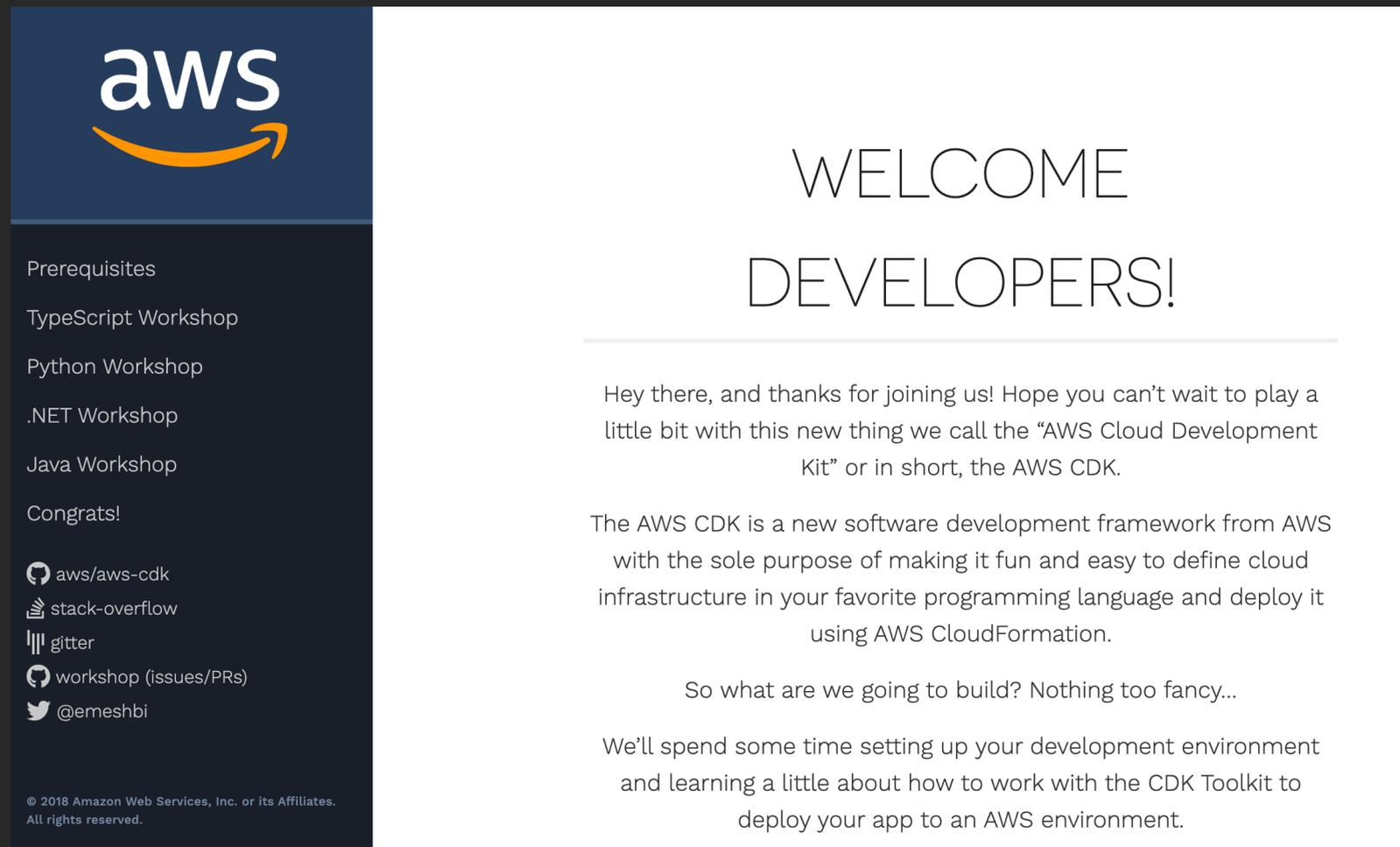
- "npm install -g aws-cdk"

3. 各言語の開発環境を用意 (例: TypeScript)

- お好きなエディタ (例: Visual Studio Code)

CDK Workshop

本日の手順のベースとなるワークショップです。今回のワークショップでは、翻訳されたものを提供しております。



aws

Prerequisites

TypeScript Workshop

Python Workshop

.NET Workshop

Java Workshop

Congrats!

 [aws/aws-cdk](#)

 [stack-overflow](#)

 [gitter](#)

 [workshop \(issues/PRs\)](#)

 [@emeshbi](#)

© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

WELCOME DEVELOPERS!

Hey there, and thanks for joining us! Hope you can't wait to play a little bit with this new thing we call the "AWS Cloud Development Kit" or in short, the AWS CDK.

The AWS CDK is a new software development framework from AWS with the sole purpose of making it fun and easy to define cloud infrastructure in your favorite programming language and deploy it using AWS CloudFormation.

So what are we going to build? Nothing too fancy...

We'll spend some time setting up your development environment and learning a little about how to work with the CDK Toolkit to deploy your app to an AWS environment.

<https://cdkworkshop.com/>

講師の画面を見ながら作業をしてください

本セクションのまとめ

- Lambda、API Gateway、CloudFormationの概要
- CDKのインストール

本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備

2. AWS CDKの紹介とハンズオン

1. AWS CDKの概要

2. AWS CDKのハンズオン ①新しいプロジェクトの作成

3. AWS CDKのハンズオン ②Hello, CDK!

4. AWS CDKのハンズオン ③Costructの補足

3. 本コースのまとめ、リソース削除

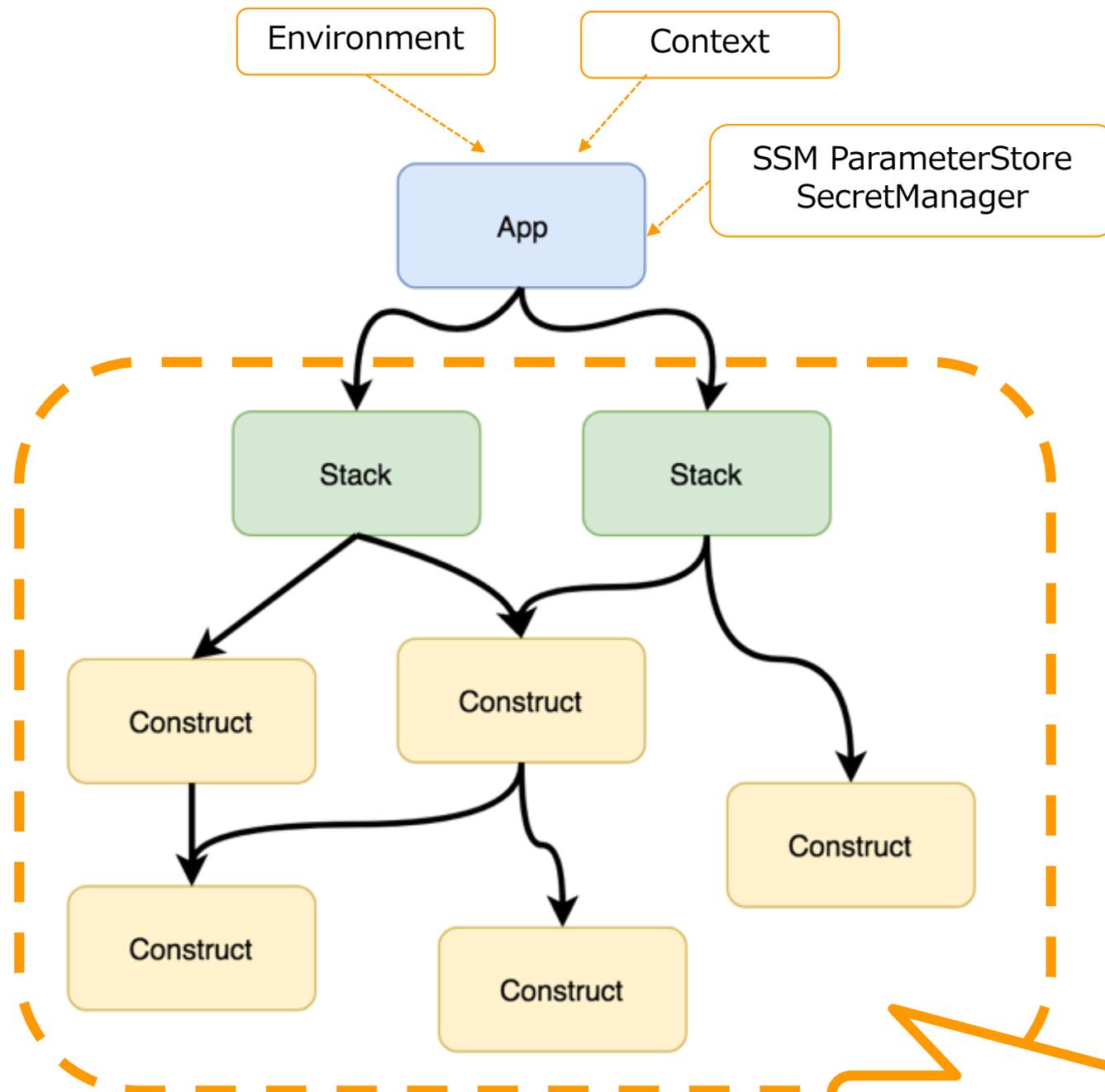
AWS CDKの概要

AWS CDK

AWSの環境を 一般のプログラミング言語で記述できるツールキット

- ソースコードからCloudFormationテンプレートを生成
- AWSのベストプラクティスが定義されたライブラリ (Construct) によって少ないコードで記述可能
- オープンソースで開発されておりユーザが拡張可能
- 2018年8月 から Developer Preview
2019年7月11日 TypeScript, Pythonが 一般利用可能 (GA) に
2019年11月25日 Java, .NETがGA

CDK アプリケーションの構成



次のスライドのオレンジ枠と比較

App

- CloudFormationテンプレートの生成とデプロイに利用する最上位要素
- 複数のStackとその依存関係を定義

Stack

- CloudFormation Stack に該当しデプロイ可能な最小単位
- リージョンとアカウントを保持

Construct

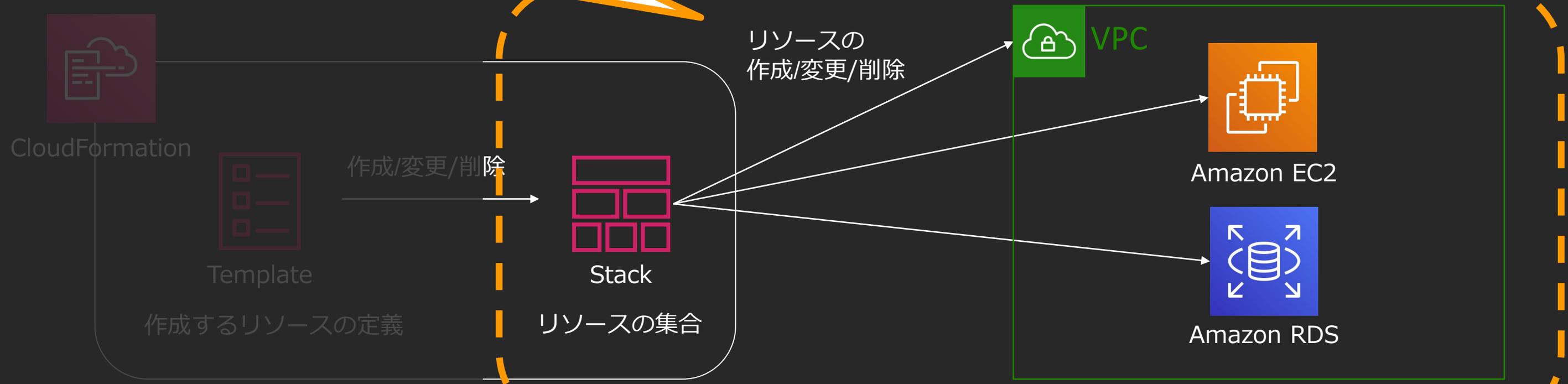
- Stackに作成されるAWSリソース
- 標準でAWS Construct Libraryを提供
- 独自に定義したり配布することが可能

AWS CloudFormation

- シンプルなテキストファイル（テンプレート）を使用して、リージョンとアカウントでアプリケーションに必要なとされるAWSリソースをプロビジョニングできるサービス
- プロビジョニングされたAWSリソースに対する変更/削除も可能で、ソフトウェアに対して行うのと同じ方法でAWSインフラを管理できる

前のスライドのオレンジ枠と比較

（AWSリソース分の料金のみ発生）



Construct

AWS CDK アプリの基本ビルディングブロック

- クラウドコンポーネントを表し、CloudFormationがリソースを作成するために必要なすべてをカプセル化
- 単一のリソースを表現したり（Amazon S3 Bucketなど）、複数のAWS CDKリソースをまたがるハイレベル コンポーネントを表現することも可能
- 標準で提供する AWS Constructs Libraryのほか、既存Constructを継承して独自のConstructを定義したり作成したConstructをパッケージとして配布することが可能

本セクションのまとめ

- CDKの概要
- CDKアプリケーションの構成
- Constructの概要

本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. **AWS CDKのハンズオン ①新しいプロジェクトの作成**
 3. AWS CDKのハンズオン ②Hello, CDK!
 4. AWS CDKのハンズオン ③Costructの補足
3. 本コースのまとめ、リソース削除

AWS CDKのハンズオン

①新しいプロジェクトの作成

CDK 開発環境の準備

1. 初期コードを生成

- `"cdk init app --language=typescript"`
- TypeScriptではないプログラミング言語を指定する場合は、`cdk init help` コマンドを実行し、指定できる引数を確認する。
- **サンプルコード付き → `"cdk init sample-app --language=typescript"`**

2. Let's coding!

CDK コードのデプロイ

1. CDKデプロイ管理用の環境 (S3バケット) を作成 (初回のみ)

- "cdk bootstrap"

2. ビルドとCloudFormationテンプレートの生成

- ビルド - "npm run build など"
- "cdk synth mystack"

3. デプロイ

- "cdk deploy mystack"

講師の画面を見ながら作業をしてください

本セクションのまとめ

- CDKの新しいプロジェクトの作成方法
- CDKのデプロイ方法
- CDKのプロジェクト構成

本コースのアジェンダ

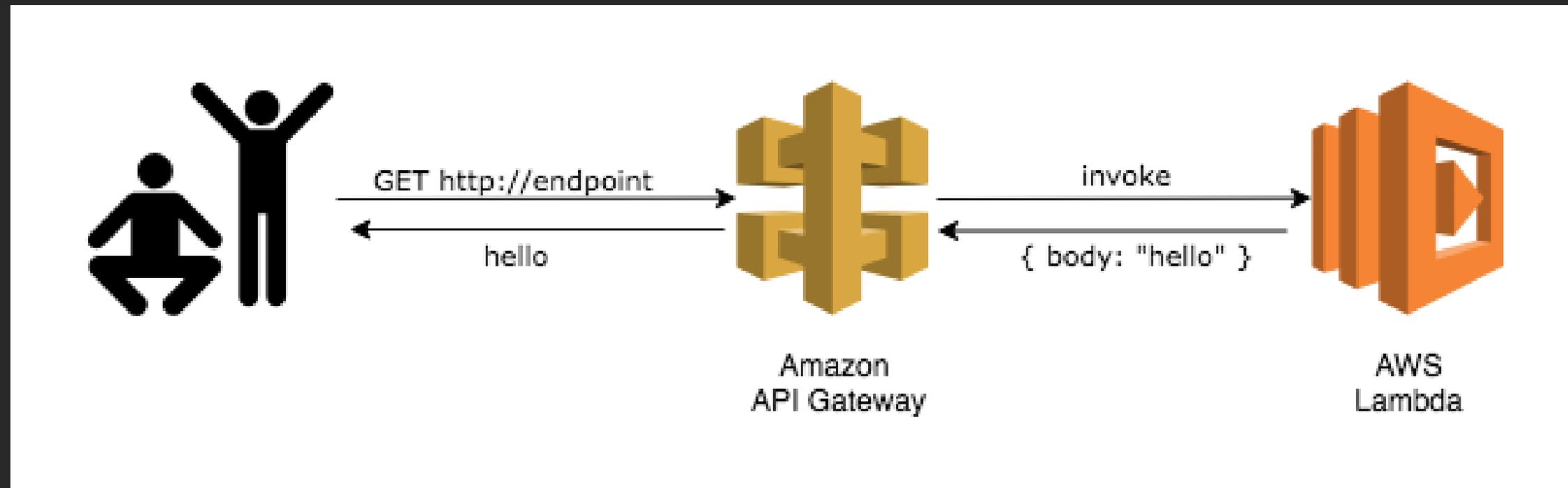
1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. AWS CDKのハンズオン ①新しいプロジェクトの作成
 - 3. AWS CDKのハンズオン ②Hello, CDK!**
 4. AWS CDKのハンズオン ③Costructの補足
3. 本コースのまとめ、リソース削除

AWS CDKのハンズオン

②Hello, CDK!

アーキテクチャ

作成したサンプルプロジェクトを整理し、以下のアーキテクチャを作成していきます。



講師の画面を見ながら作業をしてください

本セクションのまとめ

- サンプルプロジェクトの整理方法
- Construct Libraryのインストール方法
- Lambdaの構築方法
- API Gatewayの構築方法

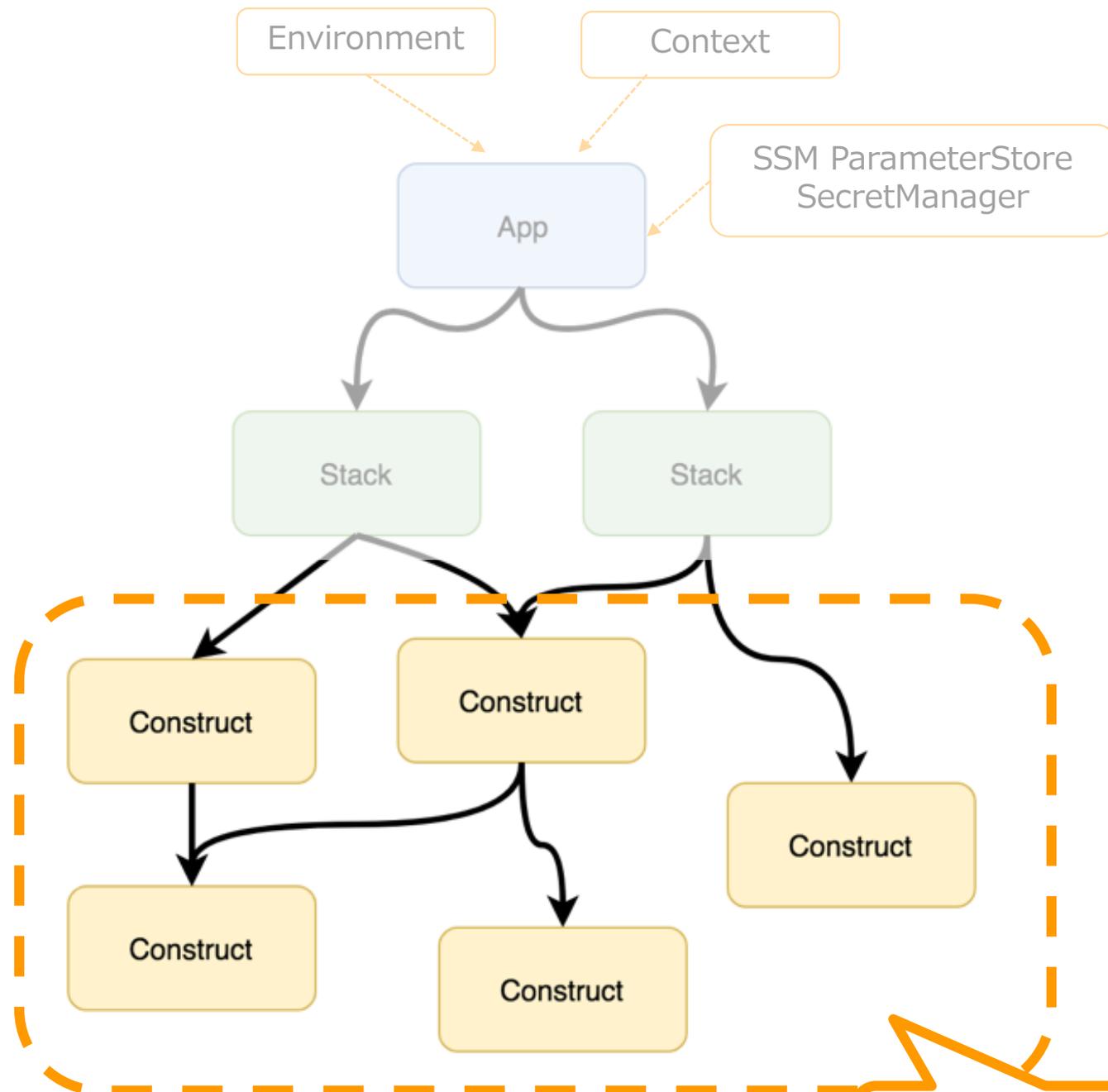
本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. AWS CDKのハンズオン ①新しいプロジェクトの作成
 3. AWS CDKのハンズオン ②Hello, CDK!
 4. **AWS CDKのハンズオン ③Costructの補足**
3. 本コースのまとめ、リソース削除

AWS CDKのハンズオン

③ Constructsの補足

CDK アプリケーションの構成



この話です

App

- CloudFormationテンプレートの生成とデプロイに利用する最上位要素
- 複数のStackとその依存関係を定義

Stack

- CloudFormation Stack に該当しデプロイ可能な最小単位
- リージョンとアカウントを保持

Construct

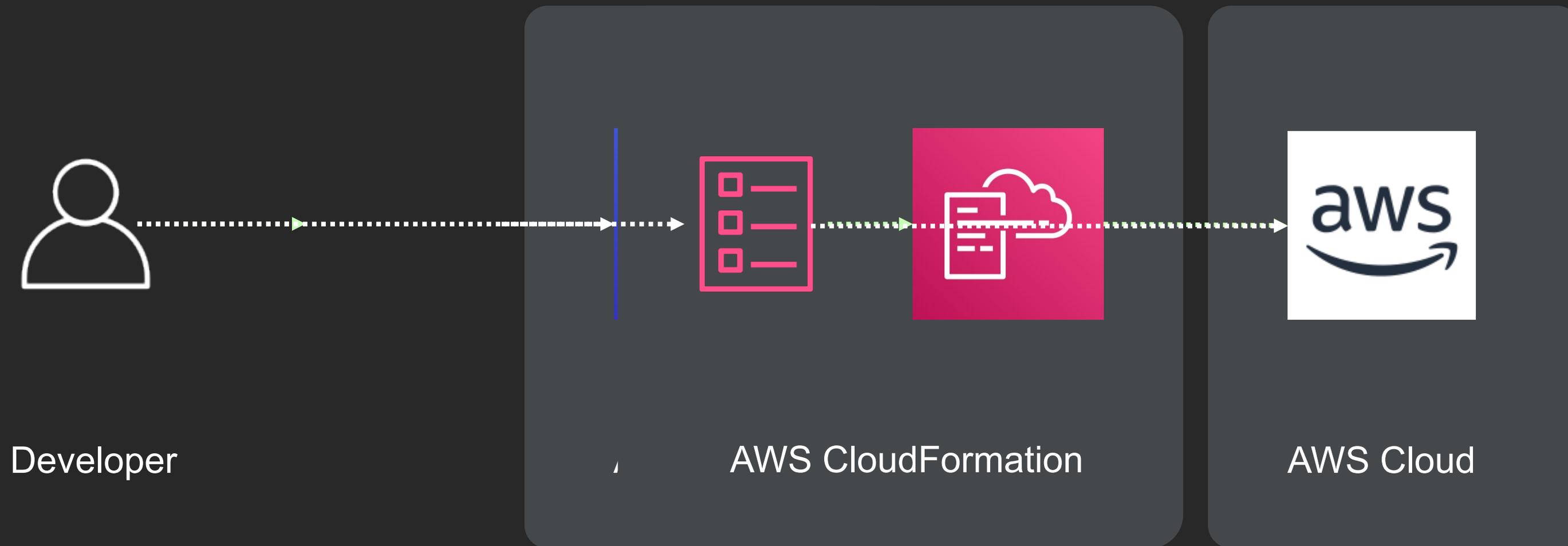
- Stackに作成されるAWSリソース
- 標準でAWS Construct Libraryを提供
- 独自に定義したり配布することが可能

AWS Constructs Library

AWS CDK が標準で提供する Construct のライブラリ

- High-level constructs (L2)
 - デフォルト値や便利なメソッドを定義したAWSリソースを表すクラス
 - 例) クラス s3 は メソッド `s3.Bucket.addLifecycleRule()` を持つ
- Low-level constructs (L1)
 - CloudFormationリソースおよびプロパティと1:1で対応 (自動生成される)
 - CfnXXXという名前 (例: `s3.CfnBucket` は `AWS::S3::Bucket` を意味)
 - すべてのプロパティを明示的に設定する必要がある
- Patterns
 - 複数のリソースを含む一般的な構成パターンを事前に定義したもの
 - `aws-ecs-patterns.LoadBalancedFargateService` など

AWS Construct Library



Resources:

MyBucket:

Type: `AWS::S3::Bucket`

MyUser:

Type: `AWS::IAM::User`

MyUserPolicy:

Type: `AWS::IAM::Policy`

Properties:

PolicyDocument:

Statement:

- Action:

- `s3:GetObject*`
- `s3:GetBucket*`
- `s3:List*`

Effect: `Allow`

Resource:

- `Fn::GetAtt: [MyBucket, Arn]`
- `Fn::Sub: "${MyBucket.Arn}/*"`

Version: `"2012-10-17"`

PolicyName: `MyUserPolicy`

Users:

- `Ref: MyUser`

AWS CloudFormation
resources



AWS CloudFormation
template language



AWS CloudFormation
resources

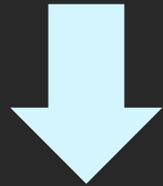
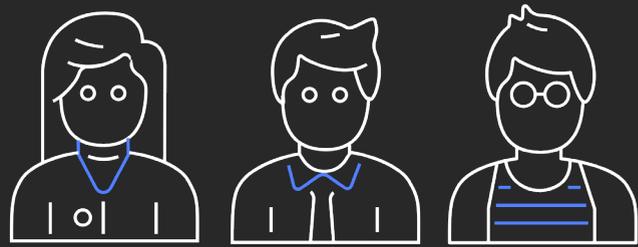


"L1"

AWS CloudFormation
template language



```
const bucket = new CfnBucket(this, 'MyBucket');
const user = new CfnUser(this, 'MyUser');
new CfnPolicy(this, 'MyUserPolicy', {
  policyName: 'MyUserPolicy',
  policyDocument: new PolicyDocument({
    statements: [new PolicyStatement({
      actions: [
        's3:GetObject*',
        's3:GetBucket*',
        's3:List*'],
      resources: [
        bucket.bucketArn,
        `${bucket.bucketArn}/*` ]
    })]
  }),
  users: [user.userName],
});
```



AWS Construct Library



"L2"

AWS CloudFormation
resources



"L1"

AWS CloudFormation
template language



```
const bucket = new s3.Bucket(this, 'MyBucket');  
const user = new iam.User(this, 'MyUser');  
  
bucket.grantRead(user);
```

ご興味のある方は、引き続きWorkshopの手順を進めてください

本コースのアジェンダ

1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. AWS CDKのハンズオン ①新しいプロジェクトの作成
 3. AWS CDKのハンズオン ②Hello, CDK!
 4. AWS CDKのハンズオン ③Costructの補足
- 3. 本コースのまとめ、リソース削除**

本コースのまとめ、リソース削除

本コースのまとめ

1. ハンズオンで利用するサービス紹介と事前準備
2. AWS CDKの紹介とハンズオン
 1. AWS CDKの概要
 2. AWS CDKのハンズオン ①新しいプロジェクトの作成
 3. AWS CDKのハンズオン ②Hello, CDK!
 4. AWS CDKのハンズオン ③Costructの補足
3. 本コースのまとめ、リソース削除

リソース削除

作成したリソースを削除するために、 `cdk destroy` コマンドを実行してください

参考資料

AWS Black Belt Online Seminar – AWS Cloud Development Kit

<https://aws.amazon.com/jp/blogs/news/webinar-bb-aws-cloud-development-kit-cdk-2020/>

GitHub

<https://github.com/aws/aws-cdk>

APIリファレンス

<https://docs.aws.amazon.com/cdk/api/latest/docs/aws-construct-library.html>

ドキュメント (Developer Guide) - 更新が早いいため英語版の参照を推奨

<https://docs.aws.amazon.com/cdk/latest/guide/home.html>

Thank you!